Software Requirements Specification

for

Semantics-based Publish/Subscribe System for Self-Configuration of Sensor Networks Services

Version 1.0

Prepared by

Mukaddim Pathan
CSIRO ICT Centre, 108 North Road, Acton, ACT 2601
20-February-2011

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Mukaddim Pathan	20.02.2011	Initial Draft	1.0

1. Introduction

1.1 Purpose

This document describes the requirements specification (SRS) for the software infrastructure (or product) that enables the self-configuration features in a sensor network deployment with the use of a semantic-based publish/subscribe component, henceforth termed as "Semantic Pub/Sub". It presents a means for a sensor network to enable autonomous structuring of contextual information and resources thus making them available to services. It will allow service customization and support employing semantic technologies. This document provides a basis for evaluating the proposal for a self-configurable sensor network service. This is the version 1.0 of the software requirements specification.

1.2 Document Conventions

When writing this SRS for Semantic Pub/Sub, the following terminologies are used:

- *Semantics*: Semantic technologies provide a machine-understandable meaning of sensed data for automatic information processing and exchange. Its use seeks to assist the integration of heterogeneous sensor resources that need to interoperate, and support the automation of service configurations.
- *Ontology*: An ontology provides a medium for capturing and reusing the knowledge and experience gained from prior efforts, it thus leads to a great level of automation at the semantic level. It can be used to describe various aspects of a sensor network service and bridge the gap between services (re)usable in various scenarios and requirements in the context of an individual client application.
- *Self-configuration*: It is a feature for sensor network services to autonomously adapt to application scenarios and context, with enhanced accessibility and reusability.
- *Publisher*: A system component that publishes sensor service configuration modeled using the W3C SSN-XG sensor ontology.
- *Subscriber*: Client applications that describes its interest as a module, i.e. managed name topics, of the ontology.
- *Registry*: By computing the ontology subsumption hierarchy with a reasoner, the registry holds a topic hierarchy of classes and properties that is made available for subscription.
- *Authoritative Entity (AE)*: The central component of the Semantic Pub/Sub system administering the operations of sensor network services.
- *Web Ontology Language (OWL)*: A knowledge representation language for authoring ontologies.
- *Abox*: It refers to an "assertion component" – a fact associated with a terminological vocabulary within a knowledge base. Grounded data about particular sensor service configurations are defined in OWL as Abox instances.
- *Tbox*: It refers to a "terminological component" – a vocabulary associated with a set of facts. Sensing concepts are defined in OWL as axioms in the Tbox.

1.3 Intended Audience and Reading Suggestions

This document is written for the researchers, software developers, advanced practitioners, documentation

writers, and users involved in sensor networks domain to initiate an open discussion for exploring development opportunities regarding self-configuration of sensor services. Section 2 discusses the steps that are to be undertaken to avail the self-configuration feature. In the next section, system features with their functional requirements are presented to highlight the major services provided by the intended product. Then the external interface requirements highlighting the logical characteristics of each interface between the software product and the users are discussed. Finally, this specification is concluded with the reference documents on which this document is based on.

1.4 Project Scope

The final product enabling the self-configuration feature will assist a sensor network service to dynamically adapt to the changing environment, including the deployment of new components or the removal of existing ones, or sudden changes in the system characteristics. These variations can happen either as random, periodic events or gradual evolutionary changes in the system. The underlying principle for building self-configuring sensor network services is to provide fundamental mechanisms upon which other networking and system services may be spontaneously specified and reconfigured. To make a sensor network service self-configurable, the following conditions must be met:

- The tasks involved in the configuration of a service are automated.
- The automation process is initiated based on situations that can be observed or detected in the sensor network system.
- An authoritative entity in the sensor network must possess sufficient knowledge of sensor network service configuration, resources, policies, and observed data.

On satisfying these conditions, it is possible to assemble the automated functions in a set of composed processes to allow a sensor network service to be self-configurable. These processes can be governed to collect necessary details from the system, analyze them to determine the required configuration changes for a sensor service, create a plan or action sequence specifying the necessary changes, and perform those actions.

1.5 References

This document builds on the following references:

- The architecture for self-configuration of sensor network services [6].
- A description of Phenonet, the application domain for the Semantic Pub/Sub product [2].
- Semantic specification of sensors [1] and use of semantics for sensor network programming [7].
- Methods for computing difference of ontology versions [3-4].
- Semantic-based service framework for query processing [5].

2. Overall Description

2.1 Product Perspective

Development of the Semantic Pub/Sub system, the product being specified in this SRS, requires to: i) define the service configuration in a standard, machine readable format; ii) publish the definition via a registry in order to make it discoverable; and iii) support a protocol to communicate with client applications. As an integral part of the product, an ontology repository provides necessary information about the underlying sensor networks in terms of observations, measurements, and sensor capabilities. The publish/subscribe interactions well adapt to the needs of self-configuration of sensor services.

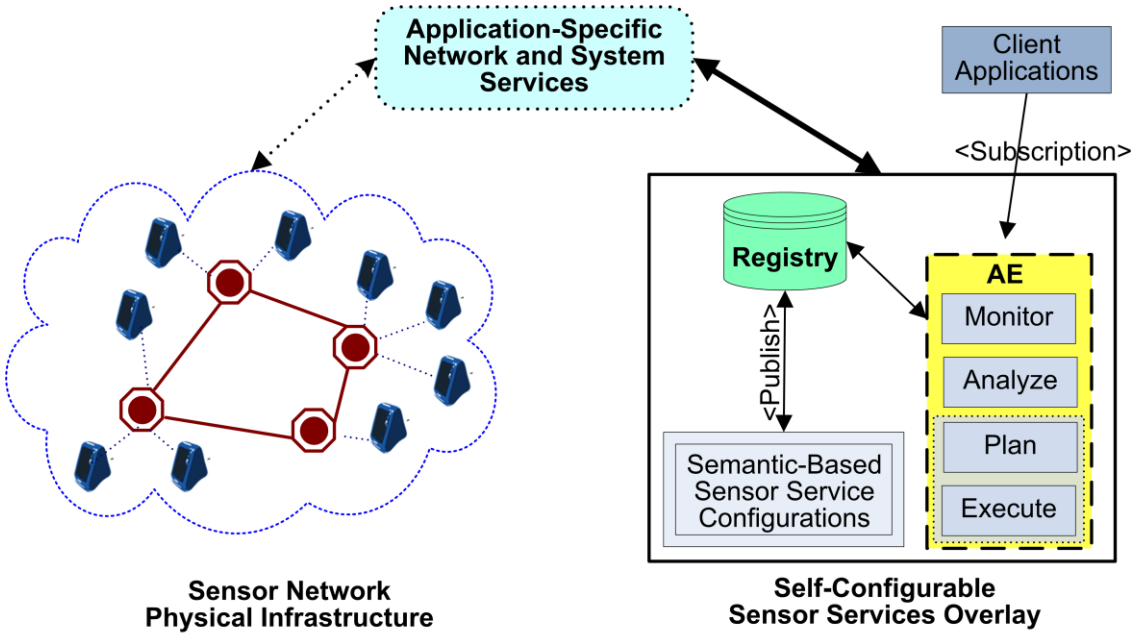

Figure 1: Abstract architecture of Semantic Pub/Sub.

Figure 1 shows an abstract architecture of the intended product. Sensor network physical layer consists of heterogeneous sensor nodes placed at distributed locations. The application-specific network and system services sit on top of the physical infrastructure to provide routing, location, naming, binding, information dissemination and aggregation services. When there are changes, within the self-configurable sensor services overlay, the runtime information is fed into the related self-configuration context ontology using a feedback mechanism. The changes trigger the execution of self-configuration logic for dynamic adaptation via monitoring, analysis, planning and execution (Figure 1). The main goal of such a mechanism is the decoupling of publishers and subscribers in time, space and synchronization.

2.2 Product Features

The Semantic Pub/Sub system can be featured with the following major lifecycle phases:

Implementation: This phase involves service design, testing and verification. The runtime infrastructure should leverage distributed components to create an optimal configuration according to both application requirements and context. It should also allow uniform access to data by applications, irrespective of the data format, source or location. An ontology can be used to represent context and sensor capabilities.

Installing and configuring: This phase involves bringing up a service to an operational state with minimal user involvement. It requires explicitly stating a user subscription to correspond to a published configuration in a formal request language. To enable this, a sensor service will entail a bootstrapping process that begins with the configurations registering itself with the Semantic Pub/Sub system.

Monitoring: This phase is included in the self-configuration management. It involves monitoring configuration changes for sensor network services. This is an essential stage for enabling self-configuration ability for a service. When coupled with event correlation or decision theory, the monitored information is useful for problem identification and recovery during system faults.

Service upgrade: Sensor network services may require upgrading them from time to time. They may subscribe/interact to an alert service that provides information on the availability of relevant upgrades and decide for themselves when to apply the upgrade.

Decommission/replacement: Alternative to an upgrade process, sensor network services for the Semantic Pub/Sub system could be implemented afresh as part of a system upgrade, removing outdated services only after the new ones obtain a proper working status.

Lifecycle management: AE performs simultaneous activities to schedule and prioritize operations. A user

interacts with a service via an appropriate interface provided by AE and retrieves the service description directly from the service. As per the node composition rule, the sensor service is configured and invoked according to application scenarios and context. During the lifetime of the system, services are dynamically reconfigured in response to sensed observations and system changes.

2.3 User Classes and Characteristics

The users of Semantic Pub/Sub system are the publishers and subscribers. Sensor network services belong to the first category who publishes service configurations expressed using the SSN-XG ontology. Subscribers are the applications from a number of clients such as scientists, plant biologists, environmental researchers, plant breeders, farmers, and funding bodies. Client applications require different services from the sensor network and they query it to retrieve data. The sensor network services may vary configuration as the result of environmental changes or other interesting phenomena. The Semantic Pub/Sub system supports this users, running in conjunction with the network and data stores, and match the right capabilities to client applications, allowing them to add and remove subscriptions dynamically as the service configurations change. In the following, example scenarios detailing user characteristics are presented.

Scenario 1: Consider a standing query (a conjunctive query over terms and properties from the ontology) submitted by a client application that is checking for temperature threshold values in a sensor network. The query represents interest in the data underlying the queried concepts. In this case, the observed properties can be sensor locations, sensor accuracy and sensor manufacturer. If any change occurs in some part of the network or the ontology itself, on which the query is built, the query will be required to change as well. A client application would thus subscribe to concepts, properties and named individuals in the query. The system then monitors change to the ontology and service configuration and notifies subscribers of changes that logically or structurally affect them. The dependence on a change can be direct or indirect (implied), since the ontology is a logical object.

Scenario 2: From the sensor network perspective, a fundamental example can be when the network is configured and programmed automatically using an ontology, with the ontology treated as data loaded into a query interface. If terms in the ontology are moved or redefined, or if additions are made either to the ontology or the capabilities of the network, the query interface and its associated compilation procedure will be notified and the reconfiguration will be reflected in the client interface to take advantage of the changes.

Scenario 3: An application service that allows the definition, and automated orchestration of new sensors and services in terms of the concepts (sensors and existing services) in the ontology can work as part of the network configuration described above or by building compositions on top of the network's capabilities. If the composite sensor application service is defined as a concept in the ontology, the concepts and roles used in the definition and construction of a new sensor can subscribe to it. If change occurs in any part of the ontology or network, on which the new sensor depends, the composition service can be notified to reconstruct the composite sensor from the updated definitions and network capabilities.

2.4 Operating Environment

The product in this SRS will be deployed within the context of the Phenonet project. The project deploys a 40-node, distributed sensor network to monitor crop growth. The sensor network consists of sensors measuring solar radiation, air temperature, soil moisture, soil temperature, and an infrared sensor measuring leaf temperature. The plant scientists involved in this project are interested in comprehensive sensor readings, based on a number of environment and water quality parameters, to run on yield and measure performance after frost, heat and drought. Based on the observed data, scientists are able to "map" microclimatic conditions such as light, temperature and soil moisture across the field to better evaluate and compare new plant varieties. By mapping these conditions and combining them with each plant's genetic profile and performance, plant scientists can de-convolve the effects of microclimate and genome, thus improving the accuracy and speed of plant breeding. Figure 2 shows an abstraction of the operating environment for the Semantic Pub/Sub system identifying the software components, assumptions and dependencies (Section 2.7).

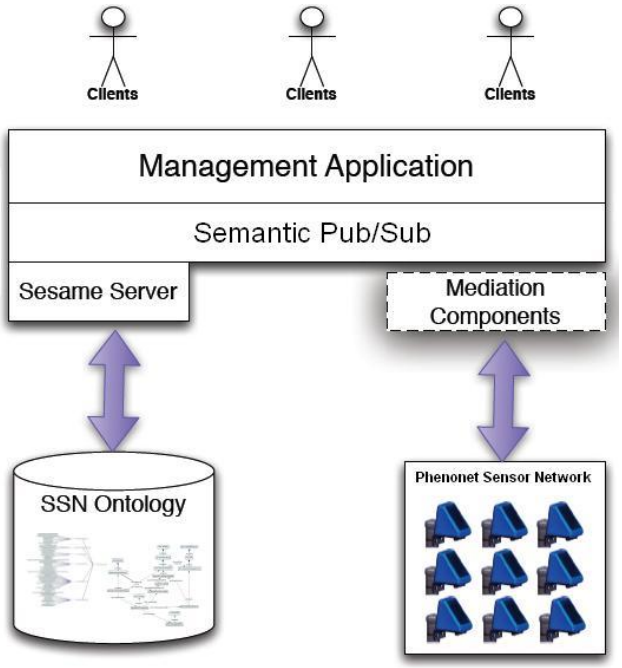

Figure 2: Operating environment.

2.5 Design and Implementation Constraints

The Semantic Pub/Sub system should formally capture the structural properties and run-time behavior of a sensor network service. The structure is defined by the service configurations and status in the system, whereas the run-time behavior is defined by the activities that execute service actions. It can be assumed that that the state of a sensor service is observable and that the results of actions influence the observables and state variables. Service types are specified in terms of one or more interfaces, whereby one interface may be attached to several configurations. An interface characterizes the behavior of a service and enables a common approach to represent the capabilities of a sensor service. Interfaces should be considered as the unit of reusability. A self-configurable service in the Semantic Pub/Sub system could be represented by a tuple **<*C, V, T*>**, where:

- ***C*** is the set of code blocks to perform service functionality $\Gamma \in \alpha \times \beta$, where α is the set of inputs and β is the set of outputs of a sensor network service and Γ defines a valid input-output set. A code block $c \in C$ makes use of state variables $v \in V$. Code blocks in the system are invoked through system operations, $t \in T$. One operation can invoke several code blocks.
- V is the set of state variables in the system, accessed through the code blocks. It can be expressed as $<\delta, \gamma>$, where δ is the set of sensors to provide services and γ is the set of constraints set that controls access to the sensor and its services. Constraints are based on state and/or context, and can control who invokes the sensor service, when and how they are invoked and configured.
- *T* is the set of operations for self-configuration that are executed by sequentially invoking code blocks in the system, based on the system state changes by manipulating the system variables in *V*.

2.6 User Documentation

Along with the software product, a user manual would be written to help people understand the working methodology and usage of the developed prototype system. It would be written for non-technical individuals and the level of content or terminology would differ considerably from, for example, a System

Administration Guide, which is more detailed and complex. The user manual would follow common user documentation styles capturing purpose and scope of the product along with key system features and operations; step-by-step instructions for using the system including conventions, messaging structures, quick references, tips for errors and malfunctions; pointers to reference documents; and glossary of terms.

2.7 Assumptions and Dependencies

This product would build on standards-based semantic technologies. In particular, W3C SSN-XG ontology will be used to represent the complete sensor network configuration, expressed in the Web Ontology Language (OWL). A significant feature of the Semantic Pub/Sub system will be its ability to support flexible subscriptions. The product will be built using an Enterprise Service Bus (ESB), such as WSO2. It provides fundamental services for complex architectures via an event-driven and standards-based messaging engine (the "bus"). The reasons for using an ESB are increased flexibility, decentralization, faster integration to existing systems, and distributed deployment. WSO2 ESB provides a built-in repository to store service configurations and metadata. Alternatively, Apache ActiveMQ, an open source message broker that implements the Java Message Service (JMS) can also be used for message exchange related to Semantic Pub/Sub. A mediation component developed in Java will integrate the Semantic Pub/Sub system with the Phenonet sensor network deployment. In addition, an open source Java framework, called Sesame will be used for storage and querying of RDF data to enable the usage of RESTful Web services APIs.

3. System Features

The major services and functional requirements for the product can be illustrated by system features. This section is organized by use cases for major system features. In the following, necessary description is provided for each use cases in the system. Each use case description provides information of the associated actors, triggering condition, preconditions, postconditions, response sequences, exceptions and functional requirements (assumptions). Being a major important section of the SRS, this section is expected to go through iterative improvement to make the most logical sense for the intended product.

3.1 Service Registration (Publish)

This feature is associated with the registration of sensor network services with the Semantic Pub/Sub system. The use case for this feature is shown in Figure 3.

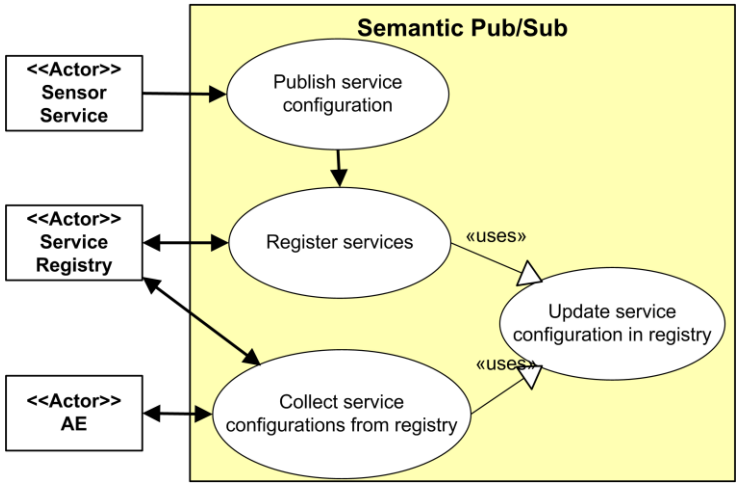

Figure 3: Sensor service registration with Semantic Pub/Sub

3.1.1 Actors

Sensor service: Publishes its configuration information to the registry.

Registry: Registers available service configurations and updates them.

Authoritative Entity (AE): Collects up-to-date service configurations from registry.

3.1.2 Trigger

Service registration is triggered when either one of the following occurs: (a) services are available as a deployed sensor network starts operating; (b) previously registered service configuration needs to be updated; and (c) new services are deployed in the system.

3.1.3 Preconditions

Available sensor network services are detected along with their configurations such as location, type, storage, data transfer rate etc.

3.1.4 Postconditions

Sensor services are registered in the registry and being updated in a regular basis.

3.1.5 Stimulus/Response Sequences

1. Each sensor network services publishes its configuration in the registry.

2. If it is a new service, its configuration is registered in the registry with a new service ID. Service ID counter is incremented.

3. Else, resource information in registry is updated in a regular basis.

4. When client application queries for service configurations, information on available services along with their IDs is supplied to the AE.

3.1.6 Exceptions

If a sensor service fails, its configuration information is removed from registry and the service ID counter is decremented.

3.1.7 Functional Requirements

REQ-1: The format for representation of service configuration is defined.

REQ-2: The interaction protocol between Sensor service-registry, registry-AE is defined.

REQ-3: There is an established norm that any service failure will be reported.

3.2 Querying of Service Configuration (Subscription)

This feature relates to the initiation request from client application to get available service configuration from the Semantic Pub/Sub system. The use case for this feature is shown in Figure 4.

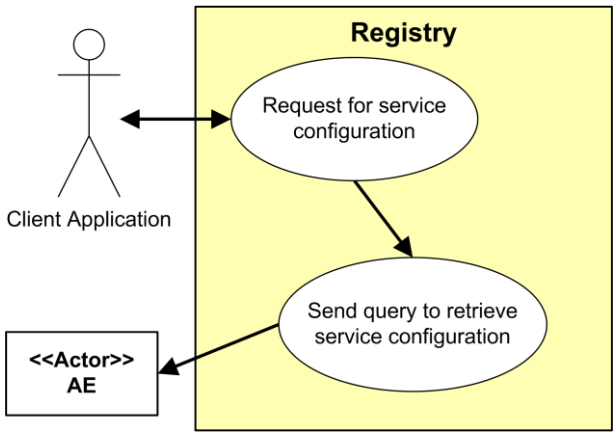

Figure 4: Client application queries to retrieve service configuration.

3.2.1 Actors

Client application: Requests for service configuration.

AE: Receives queries from client application to retrieve relevant service configurations.

3.2.2 Trigger

It is invoked when a client application wants to subscribe to the Semantic Pub/Sub system and show its interest in the form of query reference to a part of the ontology to describe available services.

3.2.3 Preconditions

Client applications generally interested, or dependent on, a portion of the complete sensor network ontology.

3.2.4 Postconditions

A query request is sent to AE to retrieve configuration data from registry.

3.2.5 Stimulus/Response Sequences

1. Client applications request for configuration data from registry.

2. AE sends query request to the registry to retrieve relevant service configurations.

3.2.6 Exceptions

If there is no matching service configuration in the registry, query outcome returns null.

3.2.7 Functional Requirements

REQ-1: Malicious queries are detected and rejected.

REQ-2: The format of the query request is defined.

3.3 Retrieval of Service Configuration (Initiation)

On receipt of the query request to retrieve configuration data, the AE invokes initialization of Pub/Sub operation. Figure 5 shows use case for this feature.

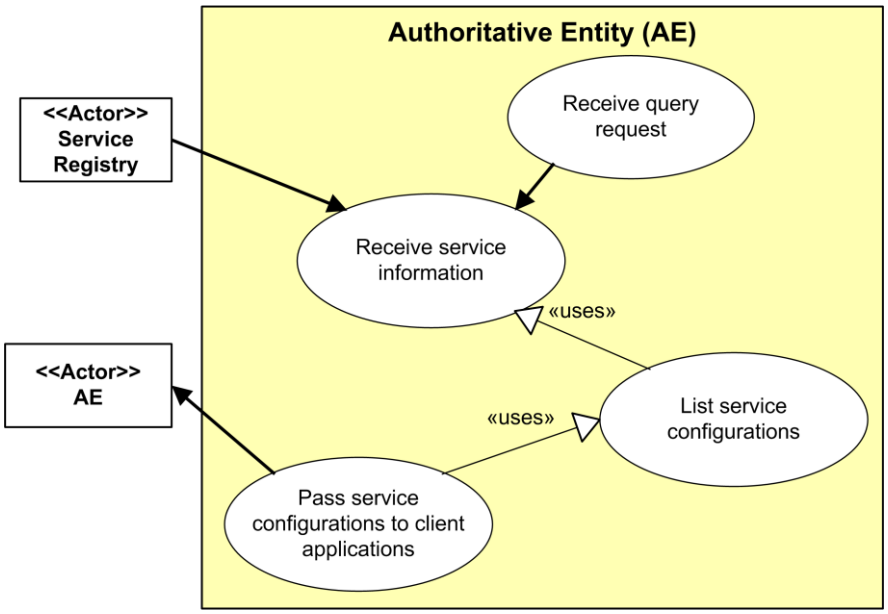

Figure 5: AE invokes initialization through providing configuration data to client application.

3.3.1 Actors

Registry: Stores service information and sends to AE for delivering to client application.

AE: Receives configuration data from registry.

3.3.2 Trigger

Upon receiving the query request to initiate semantic pub/sub operation, AE retrieves service configuration data and passes it to client application.

3.3.3 Preconditions

Query request is received.

3.3.4 Postconditions

Service configurations are retrieved and they are sent to client application.

3.3.5 Stimulus/Response Sequences

1. A query request to initialize semantic pub/sub operation is received by AE.
2. AE retrieves service information from the registry.
3. AE passes configuration data to client applications.

3.3.6 Exceptions

If the query requests cannot be accepted as there is no matching to the stored service configurations, client applications receive null.

3.3.7 Functional Requirements

REQ-1: The format of service configurations is defined.

REQ-2: AE-registry, AE-client application interaction protocols are defined.

REQ-3: AE works in conjunction with the registry to establish semantic pub/sub operations.

3.4 Adaptation to Changes (Operational Management)

This feature is responsible for adaptation of sensor network services to changing context, notifying client applications of changes and for ensuring effective operations of the Semantic Pub/Sub system. The use case for this feature is shown in Figure 6.

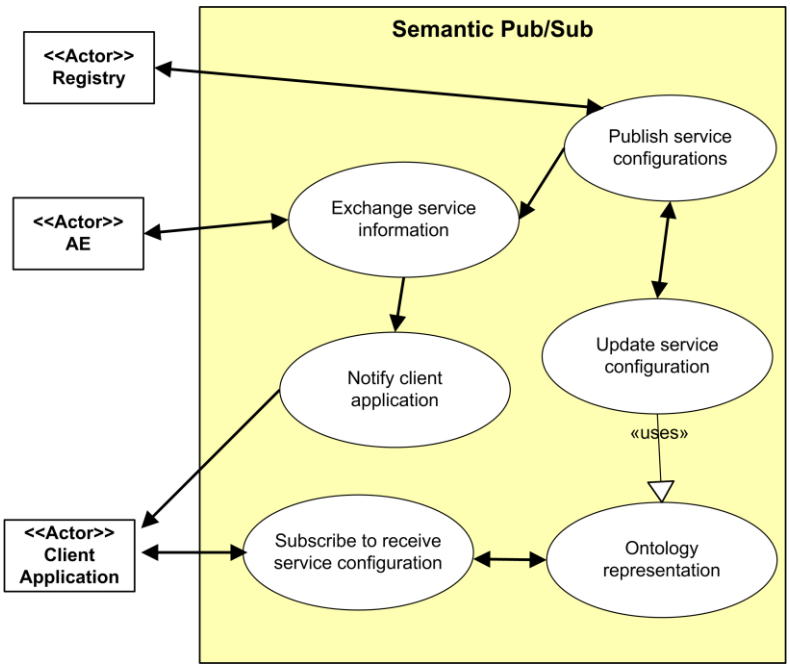

Figure 1: Operations of Semantic Pub/Sub

3.4.1 Actors

AE: Exchanges service configurations and notify client application.

Registry: Publish service configurations and update them.

Client application: Subscribes to receive service configuration and gets notified of changes.

3.4.2 Trigger

Once client application subscribes to receive updates to service configurations and get notified during the lifetime of the Semantic Pub/Sub system.

3.4.3 Preconditions

A client application has subscribed to a part of the ontological representation of available services.

3.4.4 Postconditions

Necessary functional tasks are carried out and administration for effective operations.

3.4.5 Stimulus/Response Sequences

1. Once a client application has subscribed to receive notifications on changes to service configuration by expressing interest to a part of the ontology, the semantic pub/sub operations are initiated.

2. Registry publishes available sensor services and updates them according to an ontological representation.

3. Query request from client application to retrieve the initial configuration data is directed to AE.

4. AE exchanges service information with the registry and notify client applications of ongoing changes.

5. AE assists in the administration and enforcement of the functional tasks.

3.4.6 Exceptions

If there is no client application interested in receiving notification of changes in the service configurations, the Semantic Pub/Sub system ceases operations.

3.4.7 Functional Requirements

REQ-1: Semantic Pub/Sub is operational.

REQ-2: Service configurations are published.

REQ-3: Malicious query requests are identified and acted upon.

REQ-4: Functional tasks are identified and carried out.

3.5 Decommission/Replacement

There could be circumstances under which the Semantic Pub/Sub system ceases its operation. This feature is associated with the use case shown in Figure 7.

3.5.1 Actors

AE: Governs the system operations.

3.5.2 Trigger

Termination condition holds.

3.5.3 Preconditions

There is a functional Semantic Pub/Sub system.

3.5.4 Postconditions

The Semantic Pub/Sub system is decommissioned.

3.5.5 Stimulus/Response Sequences

1. The Semantic Pub/Sub system is currently in operation and client applications are notified of changes to service configurations that are of interest to them.

2. If there is no subscribing client application, the system is decommissioned under governance of AE.

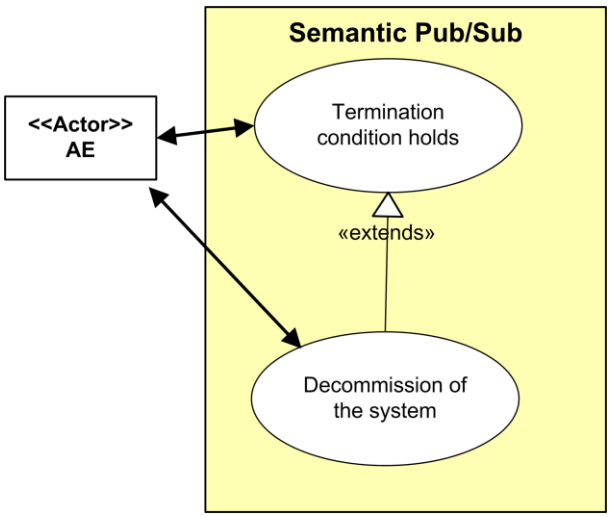

Figure 7: Decommission of Semantic Pub/Sub

3.5.6 Exceptions

If there are exceptional circumstances such as natural disaster, theft, etc. for which the deployed sensor network is affected, the physical infrastructure is restored to continue operation.

3.5.7 Functional Requirements

REQ-1: Termination conditions are identified.

REQ-2: Administrative tasks of AE are established.

4. External Interface Requirements

4.1 User Interfaces

This section describes the logical characteristics of each interface between the intended software product and the users. For user interface design, common GUI standards will be followed along with the presence of keyboard shortcuts, error message display standards etc., and standard buttons and functions (i.e. help) will appear on every screen. Details of the user interface design are intended to be documented in a separate user interface specification.

4.2 Hardware Interfaces

It is intended that the Semantic Pub/Sub system will be deployed within the context of the Phenonet project. Therefore, relevant hardware interfaces for Fleck sensors will be used to program the deployed sensor nodes. In addition, common hardware interfaces for gateway operation and communication will be used.

4.3 Software Interfaces

There will be a Web-based interface for the intended product, through which client applications can subscribe to the published service configurations and get notified. Relevant Java libraries, Java Server Faces

(JSF) or Java Server Pages (JSP), Tomcat server and MySQL databases will be used to make a complete concept demonstrator of the product. Details of the software interface will be included as the product goes through full-fledged development.

4.4 Communications Interfaces

Java Messaging Service (JMS) will be used to notify subscribing client applications. RESTful Web services APIs will be exposed to communicate with the product and integrate into a deployed sensor network. Details will be available over the development of the product.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The performance of the system will be tested under different scenarios and results will be demonstrated in terms of performance metrics such as response time, scalability, throughput etc. Detailed performance evaluation will be carried out to demonstrate superior performance of the product.

5.2 Software Quality Attributes

Additional quality characteristics of the product may also be important to the customers or developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. These additional attributes will be elaborated as the product reaches maturity.

6. References

[1] M. Compton, C. Henson, H. Neuhaus, L. Lefort, and A. Sheth, "A survey of the semantic specification of sensors," *Proc. 2nd International Workshop on Semantic Sensor Networks (SSN'09),* 2009.

[2] B. Furbank, X. Sirault, and D. Deery, "Phenonet: A distributed sensor network for field crop phenotyping," *Proc. 1st International Plant Phenomics Symposium : from Gene to Form and Function,* 2009.

[3] B. Konev, D. Walther, and F. Wolter, "The logical difference problem for description logic terminologies," *Automated Reasoning, Lecture Notes in Computer Science,* vol. 5195, pp. 259-274, 2008.

[4] R. Kontchakov, F. Wolter, and M. Zakharyaschev, "Can you tell the difference between DL-Lite ontologies," *Proc. 11th International Conference on Principles of Knowledge Representation and Reasoning,* pp. 285–295, 2008.

[5] L. Li and K. Taylor, "A framework for semantic sensor network services," *Proc. International Conference on Service Oriented Computing (ICSOC'08),* pp. 347-361, 2008.

[6] M. Pathan, K. Taylor, and M. Compton, "Semantics-based plug-and-play configuration of sensor network services," *Proc. 3rd International Workshop on Semantic Sensor Networks (SSN'10),* Springer, 2010.

[7] K. Taylor and P. Penkala, "Using Explicit Semantic Representations for User Programming of Sensor Devices," *Proc. Australasian Ontology Workshop,* 2009.